

CLAIMS

What is claimed is:

1. A method for determining an output port corresponding to the next node to which a packet is to be directed in a computer network, the method comprising:
 - (a) constructing a data structure based on a plurality of variable-length network address prefixes;
 - (b) storing the data structure in a first memory device;
 - (c) storing, in a second memory device, a set of output port identifiers corresponding to the network address prefixes;
 - (d) traversing the data structure in the first memory device based on bits in an input address to determine a location in the data structure corresponding to the longest network address prefix that matches the input address; and
 - (e) determining, based on the location in the data structure, an offset in the second memory device for the output port identifier corresponding to the input address.
2. The method of claim 1 wherein constructing a data structure includes constructing a trie data structure and storing a bit pattern indicative of the trie data structure in the first memory device.
3. The method of claim 2 wherein traversing the data structure comprises:
 - (a) initializing a start pointer to point to a first X-bit pattern in the bit pattern stored in the first memory device, the first X-bit pattern corresponding to a first level in the trie data structure, X being an integer equal to the degree of the trie data structure;

- 5 (b) reading the first $\log_2(X)$ bits of the input address and calculating an offset in the bit pattern stored in the first memory device based on the value of the first $\log_2(X)$ bits;
- (c) determining whether a bit pointed to by the offset has a first value or a second value;
- 10 (d) in response to determining that the bit has a first value, incrementing the current level of the start pointer, calculating an offset based on a number of bits having the first value before the offset in the previous level of the trie data structure, advancing the start pointer to point to a new bit position in the current level based on the number of bits having the first value, and repeating steps (b)-(d) for each $\log_2(X)$ bits in the input address until step (c) results in the offset pointing to a bit having the second value; and
- 15 (e) in response to determining that the bit pointed to by the offset has the second value, calculating the total number of bits having the first value up to and including the bit that led to the current position of the offset, wherein the total number of bits having the first value calculated corresponds to the offset in the second memory device for extracting the output port identifier.
- 20 4. The method of claim 2 wherein storing the data structure in a first memory device includes storing a single bit for each node in the trie data structure in the first memory device.
5. The method of claim 2 wherein storing the data structure in a first memory device includes omitting next-hop address pointer information from the
- 25 first memory device.

- 25

bits in an input network address includes traversing a bit pattern stored in a first memory device and wherein extracting the output port identifier comprises extracting the output port identifier from a second memory device of lower speed than the first memory device.

5 10. The method of claim 8 wherein determining a location in the trie data structure includes locating an offset with no children in the trie data based on all of the bits in the input network address.

11. The method of claim 8 wherein steps (a)-(d) are implemented in hardware.

10 12. The method of claim 8 wherein steps (a)-(d) are implemented in software.

13. The method of claim 8 wherein steps (a)-(d) are implemented in a combination of hardware and software.

14. A fast packet forwarding engine for performing fast lookups for output port identifiers corresponding to network addresses, the fast packet forwarding engine comprising:

15

(a) a first memory device storing a trie data structure corresponding to a plurality of variable-length network address prefixes, the trie data structure including a plurality of leaf nodes at locations corresponding to variable-length network address prefixes;

20 (b) a second memory device storing a plurality of output port identifiers; and

(c) a processor for determining the location of a leaf node in the trie data structure in the first memory device based on bits in an input address, and for determining an offset for extracting an output port identifier from the second memory device based on the location.

25

15. The fast packet forwarding engine of claim 14 wherein the trie data structure includes internal nodes and pointer information is not stored at the internal nodes.
16. The fast packet forwarding engine of claim 14 wherein the first memory device has a shorter access time than the second memory device.
17. The fast packet forwarding engine of claim 14 wherein the first memory device stores a bit pattern indicative of the trie data structure, wherein each node in the trie data structure is represented by a single bit in the bit pattern.
18. The first packet forwarding engine of claim 14 wherein the first memory device and the processor are located on a first chip.
19. The first packet forwarding engine of claim 18 wherein the second memory device is located on a second chip separate from the first chip.
20. The fast packet forwarding engine of claim 14 wherein the processor comprises:
 - (a) a bit extractor for extracting bits from the input address;
 - (b) a mask generator for masking bits generating a mask used to calculate a sum of bits having a first value corresponding to the offset in the second memory device; and
 - (c) an offset calculator for summing the number of bits having the first value using the mask and the bits extracted from the input address, wherein the offset corresponds to the sum.